A brief description of the HHO algorithm

1. Harris hawks optimization (HHO)

1.1. Exploration phase

In HHO, the Harris' hawks perch randomly on some locations and wait to detect a prey based on two strategies.

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q < 0.5 \end{cases} \quad (1)$$

where $X(t+1)$ is the position vector of hawks in the next iteration t, $X_{rabbit}(t)$ is the position of rabbit, $X(t)$ is the current position vector of hawks, r_1, r_2, r_3, r_4, and q are random numbers inside (0,1), which are updated in each iteration, LB and UB show the upper and lower bounds of variables, $X_{rand}(t)$ is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks. The average position of hawks is attained using Eq. (2):

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t) \quad (2)$$

where $X_i(t)$ indicates the location of each hawk in iteration t and N denotes the total number of hawks.

1.2. Transition from exploration to exploitation

To model this step, the energy of a rabbit is modeled as:

$$E = 2E_0(1 - \frac{t}{T}) \quad (3)$$

where E indicates the escaping energy of the prey, T is the maximum number of iterations, and E_0 is the initial state of its energy.

1.3. Exploitation phase

1.3.1. Soft besiege

This behavior is modeled by the following rules:

$$X(t+1) = \Delta X(t) - E \left| J X_{rabbit}(t) - X(t) \right| \quad (4)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (5)$$

where $\Delta X(t)$ is the difference between the position vector of the rabbit and the current location in iteration t, r_5 is a random number inside (0,1), and $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure. The J value changes randomly in each iteration to simulate the nature of rabbit motions.

1.3.2. Hard besiege

In this situation, the current positions are updated using Eq. (6):

$$X(t+1) = X_{rabbit}(t) - E \left| \Delta X(t) \right| \quad (6)$$

1.3.3. Soft besiege with progressive rapid dives

To perform a soft besiege, we supposed that the hawks can evaluate (decide) their next move based on the following rule in Eq. (7):

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X(t)\right| \tag{7}$$

We supposed that they will dive based on the LF-based patterns using the following rule:

$$Z = Y + S \times LF(D) \tag{8}$$

where D is the dimension of problem and S is a random vector by size $1 \times D$ and LF is the levy flight function, which is calculated using Eq. (9):

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \sigma = \left(\frac{\Gamma(1+\beta) \times sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})})}\right)^{\frac{1}{\beta}} \tag{9}$$

where u, v are random values inside (0,1), β is a default constant set to 1.5.

Hence, the final strategy for updating the positions of hawks in the soft besiege phase can be performed by Eq. (10):

$$X(t+1) = \begin{cases} Y & if F(Y) < F(X(t)) \\ Z & if F(Z) < F(X(t)) \end{cases} \tag{10}$$

where Y and Z are obtained using Eqs.(7) and (8).

1.3.4. Hard besiege with progressive rapid dives

The following rule is performed in hard besiege condition:

$$X(t+1) = \begin{cases} Y & if F(Y) < F(X(t)) \\ Z & if F(Z) < F(X(t)) \end{cases} \tag{11}$$

where Y and Z are obtained using new rules in Eqs.(12) and (13).

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X_m(t)\right| \tag{12}$$

$$Z = Y + S \times LF(D) \tag{13}$$

where $X_m(t)$ is obtained using Eq. (2).

1.4. Pseudocode of HHO

The pseudocode of the proposed HHO algorithm is reported in Algorithm 1.

References

Harris Hawks Optimization: Algorithm and Applications, Ali Asghar Heidari and Seyedali Mirjalili and Hossam Faris and Ibrahim Aljarah and Majdi Mafarja and Huiling Chen, Future Generation Computer Systems, 2019.

Algorithm 1 Pseudo-code of HHO algorithm

```
Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value
Initialize the random population X_i (i = 1, 2, ..., N)
while (stopping condition is not met) do
    Calculate the fitness values of hawks
    Set X_rabbit as the location of rabbit (best location)
    for (each hawk (X_i)) do
        Update the initial energy E_0 and jump strength J      ▷ E0=2rand()-1, J=2(1-rand())
        Update the E using Eq. (3)
        if (|E| ≥ 1) then                                       ▷ Exploration phase
            Update the location vector using Eq. (1)
        if (|E| < 1) then                                       ▷ Exploitation phase
            if (r ≥0.5 and |E| ≥ 0.5 ) then                     ▷ Soft besiege
                Update the location vector using Eq. (4)
            else if (r ≥0.5 and |E| < 0.5 ) then                ▷ Hard besiege
                Update the location vector using Eq. (6)
            else if (r <0.5 and |E| ≥ 0.5 ) then                ▷ Soft besiege with progressive rapid dives
                Update the location vector using Eq. (10)
            else if (r <0.5 and |E| < 0.5 ) then                ▷ Hard besiege with progressive rapid dives
                Update the location vector using Eq. (11)
Return X_rabbit
```